Solving Interpretable Kernel Dimension Reduction

Chieh Wu, Jared Miller, Yale Chang, Mario Sznaier, Jennifer G. Dy
Dept .of Electrical and Computer Engineering, Northeastern University

Northeastern University

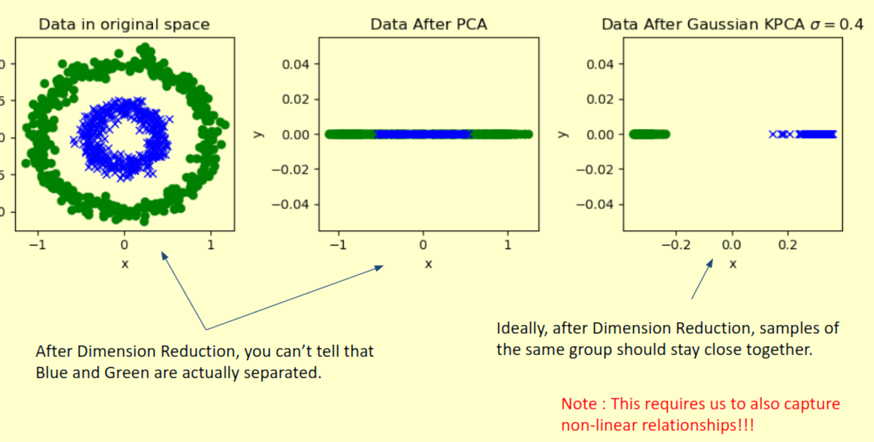
What is IKDR?

Principal Component Analysis (PCA) is the most commonly used Dimension Reduction (DR) technique. It is also an **interpretable** way to reduce the dimension.

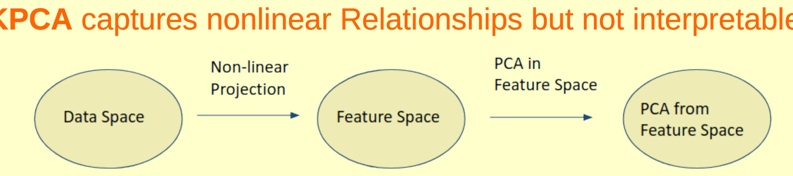
We know exactly how the new features relate to the original features.

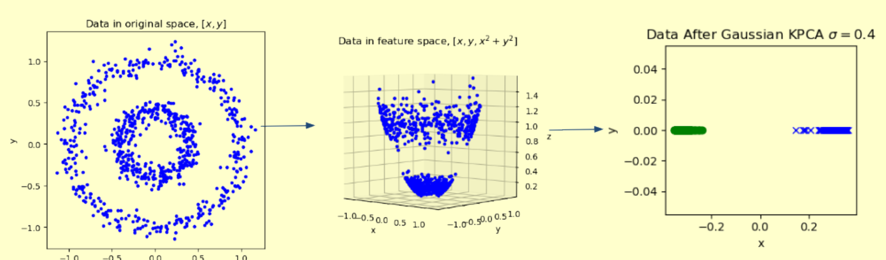
$$\begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} x_1 w_{11} + x_2 w_{12} + x_3 w_{13} \\ x_1 w_{21} + x_2 w_{22} + x_3 w_{23} \end{bmatrix}$$

But PCA cannot capture **nonlinear Relationships.**



KPCA captures nonlinear Relationships but not interpretable.





KPCA is very powerful, but …….

Problem 1: It does not use labels to guide the dimension reduction.

Problem 2: Since KPCA is PCA in the feature space, it's not obvious what they mean.

Here is the Gaussian Kernel feature map:

$$\phi(x) = e^{-x^2/2\sigma^2}\left[1, \sqrt{\frac{1}{1!\sigma^2}}x, \sqrt{\frac{1}{2!\sigma^4}}x^2, \sqrt{\frac{1}{3!\sigma^6}}x^3, \dots\right]^T$$

Not too obvious what running PCA on these features mean.

Interpretable Kernel Dimension Reduction (IKDR) solves both problems…

How IKDR produce interpretable results.

KPCA → Φ(X) W — Finds the W matrix in Feature Space that keeps the most information.

Interpretable KDR → Φ(XW) — Finds the W matrix in **original data space** that keeps the most information in the Feature Space.

HSIC is a general objective for capturing non-linear dependence to achieve IKDR.

$$\max_W HSIC(XW, Y) \quad \text{s.t } W^TW = I$$

HSIC(X,Y) measures the non-linear dependence between X and Y in Feature Space.

Although this make the solution interpretable, it is very difficult to solve.

In general, many IKDR problems have a common objective.

$$\max_W \sum_{i,j} \Gamma_{i,j} K_{XW_{i,j}} \quad \text{s.t } W^TW = I \qquad (1)$$

Symmetric Positive Definite Matrix — Kernel Matrix on XW — Grassmann Manifold Constraint.

Where is IKDR used?

Supervised Dimension Reduction for Classification

$$\max_W \quad HSIC(XW, Y) \quad \text{s.t} \quad W^TW = I$$

Unsupervised Dimension Reduction for Clustering

$$\max_{W,Y} \quad HSIC(XW, Y) \quad \text{s.t} \quad W^TW = I$$

Semi-supervised Dimension Reduction for Clustering Using Multiple Expert Sources

$$\max_{W,Y} \quad \text{Tr}(Y^T \mathcal{L}_W Y) + \mu \text{Tr}(K_{XW} H K_{\hat{Y}} H)$$

$$\text{s.t} \quad \mathcal{L}_W = D^{-\frac{1}{2}} K_{XW} D^{-\frac{1}{2}} W^TW = I, Y^TY = I$$

Alternative Clustering via Dimension Reduction

$$\max_{W,Y} \quad \text{Tr}(K_{XW} H K_Y H) - \mu \text{Tr}(K_{XW} H K_{\hat{Y}} H)$$

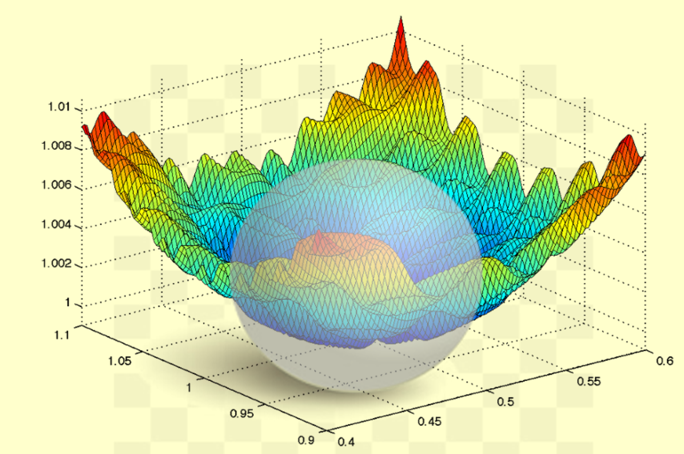
$$\text{s.t} \quad W^TW = I, Y^TY = I$$

Publications that used IKDR

- Barshan, Elnaz, et al. "Supervised principal component analysis: Visualization, classification and regression on subspaces and submanifolds." Pattern Recognition 44.7 (2011): 1357-1371.
- Masaeli, Mahdokht, Jennifer G. Dy, and Glenn M. Fung. "From transformation-based dimensionality reduction to feature selection." Proceedings of the 27th International Conference on Machine Learning (ICML-10). 2010.
- Niu, Donglin, Jennifer G. Dy, and Michael I. Jordan. "Multiple non-redundant spectral clustering views." Proceedings of the 27th international conference on machine learning (ICML-10). 2010.
- Niu, Donglin, Jennifer Dy, and Michael I. Jordan. "Dimensionality reduction for spectral clustering." Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics. 2011.
- Wu, Chieh, et al. "Iterative spectral method for alternative clustering." International Conference on Artificial Intelligence and Statistics. 2018.
- Chang, Yale, et al. "Clustering with Domain-Specific Usefulness Scores." Proceedings of the 2017 SIAM International Conference on Data Mining. Society for Industrial and Applied Mathematics, 2017.

Why is IKDR Difficult?

Optimizing W is highly non-convex and the solution must intersect the Stiefel Manifold



$$\max_W \sum_{i,j}^{n} \Gamma_{i,j} e^{-\frac{W^T(x_i - x_j)(x_i - x_j)^T W}{2\sigma^2}}$$

$$\text{s.t} \quad W^TW = I$$

Existing Solutions
- Dimension Growth
- Optimization Via Stiefel Manifold.
- Optimization Via Grassmann Manifold.
- Stochastic Gradient Descent.

Problems with Existing Solutions
- very slow
- Difficult to implement
- stuck at saddle point
- poor results

Our Solution
- The Iterative Spectral Method (ISM)

Our Solution : The Iterative Spectral Method (ISM)

We identified a special family of kernels (The ISM family) with the following properties:

1. Each kernel within the family has an associated scaled covariance matrix Φ.
2. The most dominant eigenvectors of Φ is the solution to Eq. (1).
3. The conic combination of ISM kernels is still in the ISM family.
4. The conic combination of Φs is the associated scaled covariance matrix for the conic combination of kernels.
5. If Φ is a function of W, then Φ can be approximated using the 2nd order Taylor series

Formal Definition of the ISM family:

Definition 1. *Given $\beta = a(x_i, x_j)^T W W^T b(x_i, x_j)$ with $a(x_i, x_j)$ and $b(x_i, x_j)$ as functions of x_i and x_j, any twice differentiable kernel that can be written in terms of $f(\beta)$ while retaining its symmetric positive semi-definite property is an ISM kernel belonging to the ISM family with an associated Φ matrix defined as*

$$\Phi = \frac{1}{2} \sum_{i,j} \Gamma_{i,j} f'(\beta) A_{i,j}. \qquad (6)$$

where $A_{i,j} = b(x_i, x_j) a(x_i, x_j)^T + a(x_i, x_j) b(x_i, x_j)^T$.

Theorem 3. *For any kernel within the ISM family, a Φ independent of W can be approximated with*

$$\Phi \approx \text{sign}(\nabla_\beta f(0)) \sum_{i,j} \Gamma_{i,j} A_{i,j}. \qquad (7)$$

The ISM Algorithm:

Algorithm 1 ISM Algorithm

Input : Data X, kernel, Subspace Dimension q
Output : Projected subspace W
Initialization : Initialize Φ_0 using Table 1.
Set W_0 to $V_{\max}$ of Φ_0.
while $||\Lambda_i - \Lambda_{i-1}||_2 / ||\Lambda_i||_2 < \delta$ **do**
  Compute Φ using Table 2
  Set W_k to $V_{\max}$ of Φ
end

Examples of Approximations of Φs

Kernel	Approximation of Φs
Linear	$\Phi_0 = X^T \Gamma X$
Squared	$\Phi_0 = X^T \mathcal{L}_\Gamma X$
Polynomial	$\Phi_0 = X^T \Gamma X$
Gaussian	$\Phi_0 = -X^T \mathcal{L}_\Gamma X$
Multiquadratic	$\Phi_0 = X^T \mathcal{L}_\Gamma X$

Table 1: Equations for the approximate Φs for the common kernels.

Examples of of Φs

Kernel	Φ Equations
Linear	$\Phi = X^T \Gamma X$
Squared	$\Phi = X^T \mathcal{L}_\Gamma X$
Polynomial	$\Phi = X^T \Psi X \quad , \quad \Psi = \Gamma \odot K_{XW,p-1}$
Gaussian	$\Phi = -X^T \mathcal{L}_\Psi X \text{ , } \Psi = \Gamma \odot K_{XW}$
Multiquadratic	$\Phi = X^T \mathcal{L}_\Psi X \text{ , } \Psi = \Gamma \odot K_{XW}^{(-1)}$

Table 2: Equations for Φs for the common kernels.

How K(x,x') become ƒ(β):

Kernel Name	$f(\beta)$	$a(x_i, x_j)$	$b(x_i, x_j)$
Linear	β	x_i	x_j
Squared	β	$x_i - x_j$	$x_i - x_j$
Polynomial	$(\beta + c)^p$	x_i	x_j
Gaussian	$e^{\frac{-\beta}{2\sigma^2}}$	$x_i - x_j$	$x_i - x_j$
Multiquadratic	$\sqrt{\beta + c^2}$	$x_i - x_j$	$x_i - x_j$

Table 3: Converting common kernels to $f(\beta)$.

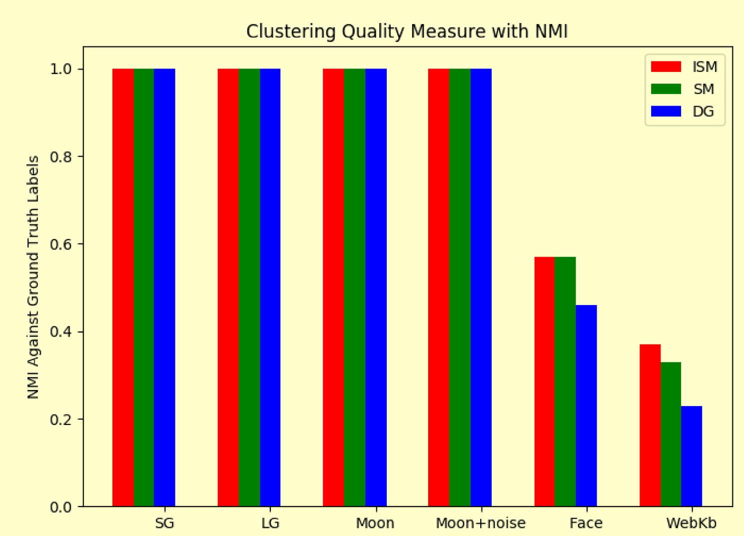
Experimental Results

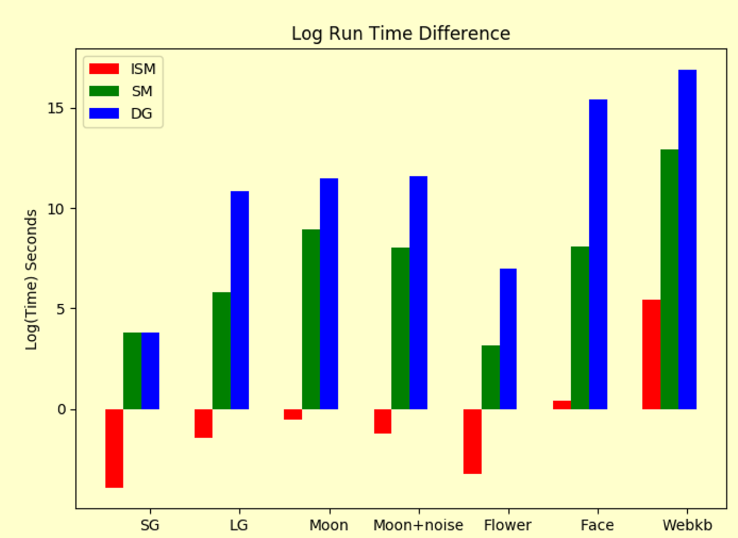
Supervised		Gaussian ISM	Gaussian DG	Gaussian SM	Gaussian GM	polynomial ISM	polynomial DG	polynomial SM	polynomial GM
Wine	Time	**0.02s ± 0.01s**	7.9s ± 2.9s	1.7s ± 0.7s	16.8m ± 3.4s	**0.02s ± 0.0s**	13.2s ± 6.2s	14.77s ± 0.6s	16.82m ± 3.6s
	Cost	**-1311 ± 26**	-1201 ± 25	-1310 ± 26	-1307 ± 25	**-114608 ± 1752**	-112440 ± 1719	-111339 ± 1652	-108892 ± 1590
	Accuracy	**95.0% ± 5%**	93.2% ± 5.5%	**95% ± 4.2%**	**95% ± 6%**	**97.2% ± 3.7%**	93.8% ± 3.9%	96.6% ± 3.7%	96.6% ± 2.7%
Cancer	Time	**0.08s ± 0.0s**	4.5m ± 103s	17s ± 12s	17.8m ± 80s	**0.13s ± 0.0s**	4m ± 1.2m	3.3m ± 3s	17.5m ± 1.1m
	Cost	**-32249 ± 338**	-30302 ± 2297	-31996 ± 499	-30998 ± 560	**-1894 ± 47**	-1882 ± 47	-1737 ± 84	-1690 ± 108
	Accuracy	97.3%± 0.3%	97.3%± 0.3%	97.3%± 0.2%	**97.4%± 0.4%**	**97.4%± 0.3%**	97.3% ± 0.3%	**97.4% ± 0.3%**	97.3% ± 0.3%
Face	Time	**0.99s ± 0.1s**	1.92d ± 11h	10s ± 5s	22.7m ± 18s	**0.7s ± 0.03s**	2.1d ± 13.9h	5.0m ± 5.7s	21.5m ± 9.8s
	Cost	**-3754 ± 31**	-3431 ± 32	-3749 ± 33	-771 ± 28	**-82407 ± 1670**	-78845 ± 1503	-37907 ± 15958	-3257 ± 517
	Accuracy	**100% ± 0%**	**100% ± 0%**	**100% ± 0%**	99.2% ± 0.2%	**100% ± 0%**	**100% ± 0%**	**100% ± 0%**	99.8% ± 0.2%
MNIST	Time	**13.8s ± 2.3s**	> 3d	2.5m ± 1.0s	> 3d	**12.1s ± 1.4s**	> 3d	2.1m ± 3s	> 3d
	Cost	**-639 ± 2.3**	N/A	-621 ± 5.1	N/A	**-639 ± 2**	N/A	-620 ± 5.1	N/A
	Accuracy	**99% ± 0%**	N/A	98.5% ± 0.4%	N/A	**99% ± 0%**	N/A	**99% ± 0%**	N/A
Unsupervised									
Wine	Time	**0.01s**	9.9s	0.6s	16.7m	**0.02s**	14.4s	2.9s	33.5m
	Cost	**-27.4**	-25.2	-27.3	-27.3	**-1600**	-1582	-1598	-1496
	NMI	**0.86**	**0.86**	**0.86**	**0.86**	**0.84**	**0.84**	**0.84**	0.83
Cancer	Time	**0.57s**	4.3m	3.9s	44m	**0.5s**	8.0m	8.8m	41m
	Cost	**-243**	-133	-146	-142	**-15804**	-14094	-15749	-11985
	NMI	**0.8**	0.79	**0.8**	0.79	0.79	**0.80**	0.79	**0.80**
Face	Time	**0.3s**	1.3d	5.3s	55.9m	**1.0s**	> 3d	22m	1.6d
	Cost	**-169.3**	-167.7	-168.9	-37	**-368**	NA	-348	-321
	NMI	0.94	**0.95**	0.93	0.89	**0.94**	N/A	0.89	0.89
MNIST	Time	**1.8h**	> 3d	1.3d	> 3d	**8.3m**	> 3d	0.9d	> 3d
	Cost	**-2105**	N/A	-2001	N/A	**-51358**	N/A	-51129	N/A
	NMI	**0.47**	N/A	0.46	N/A	**0.32**	N/A	**0.32**	N/A

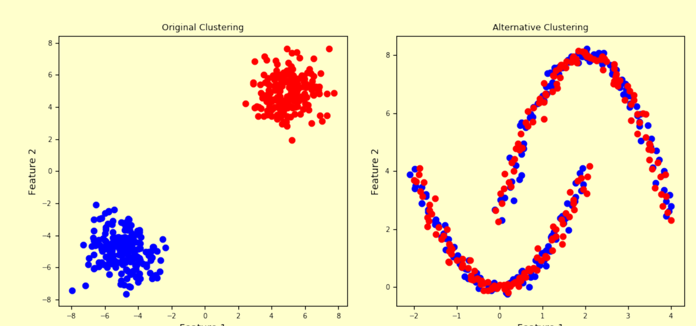
Table 4: Run-time, cost, and objective performance are recorded under supervised/unsupervised objectives. ISM is significantly faster compared to other optimization techniques while achieving lower objective cost.

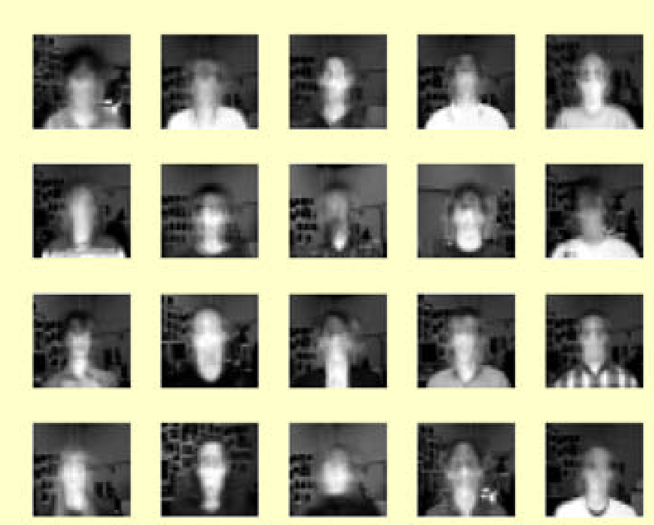
		Supervised Linear	Supervised Squared	Supervised Multiquad	Supervised G+P	Unsupervised	Unsupervised Linear	Unsupervised Squared	Unsupervised Multiquad
Wine	Time	**0.003s ± 0s**	0.01s ± 0s	0.02s ± 0.01s	0.007s ± 0s	Time	**0.02s**	0.04s	0.06s
	Accuracy	97.2% ± 2.8%	96.6% ± 3.7%	97.2% ± 3.7%	**98.3% ± 2.6%**	NMI	0.85	0.85	**0.88**
Cancer	Time	**0.02s ± 0.002s**	0.09s ± 0.02s	0.15s ± 0.01s	0.06s ± 0.004s	Time	**0.23s**	0.5s	0.56s
	Accuracy	97.2% ± 0.3%	97.3% ± 0.04%	**97.4% ± 0.003%**	**97.4% ± 0.003%**	NMI	0.80	0.79	**0.84**
Face	Time	**0.2s ± 0.2s**	0.3s ± 0.2s	0.3s ± 0.2s	0.5s ± 0.03s	Time	**0.68s**	0.92s	3.7s
	Accuracy	97.3% ± 0.3%	97.1% ± 0.4%	97.3% ± 0.4%	**100% ± 0%**	NMI	0.93	**0.95**	0.92
MNIST	Time	**6.4s ± 0.4s**	17.4s ± 0.4s	10.6m ± 1.9m	17.6s ± 2.5s	Time	**3.1m**	4.7m	52m
	Accuracy	99.1% ± 0.1%	**99.3% ± 0.2%**	99.1% ± 0.1%	**99.3% ± 0.2%**	NMI	**0.54**	**0.54**	**0.54**

Table 5: Run-time and objective performance are recorded across several kernels within the ISM family. It confirms the usage of Φ or linear combination of Φ in place of kernels.









(a) Identity View (Mean Images)

(b) Pose View (Mean Images)

ISM's Theoretical Foundation

Theorem 1:
Given a full rank Φ with an eigengap as defined by Eq. (80), a fixed point W* of algorithm 1 satisfies the 2nd order necessary condition using any ISM Kernel.

$$\left(\min_i \bar{\Lambda}_i - \max_j \Lambda_j \right) \geq \mathcal{C}. \qquad (80)$$

Theorem 2:
A sequence of subspaces generated by Algorithm 1 contains a converging subsequence.

Theorem 3:
For any kernel within the ISM family, a Φ Independent of W can be approximated with

$$\Phi \approx \text{sign}(\nabla_\beta f(0)) \sum_{i,j} \Gamma_{i,j} A_{i,j}.$$

Proposition 1:
Any conic combination of ISM kernels is still an ISM kernel.

Corollary 1:
The Φ matrix associated with a conic combination of kernels is the conic combination of Φs associated with each individual kernel.

Acknowledgments: This work was made possible by the National Science Foundation (NSF IIS-1546428). The PROTECT data is supported by the National Institute of Environmental Health Sciences (P42ES017198).